

# LET'S GROW MORE

## NAME - VIVEK SHARMA

## DATA SCIENCE INTERN

### Task 2- Prediction using DecisionTree

Description:

Create the Decision Tree classifier and visualize it graphically.

The purpose is if we feed any new data to this classifier, it would be able to predict the right class accordingly.

DATASET - <https://bit.ly/3KcTdx>

### Importing Libraries</span>

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as ss
ss.set(color_codes=True)
%matplotlib inline

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn.tree import plot_tree
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

In [2]: #importing data
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

In [3]: #checking for null values
df.isnull()
```

```
Out[3]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

```
In [4]: df.isnull().sum()
```

```
Out[4]:
```

sepal length (cm)	0
sepal width (cm)	0
petal length (cm)	0
petal width (cm)	0
dtype:	int64

```
In [5]: #shape of the data
df.shape
```

```
Out[5]: (150, 5)
```

```
In [6]: df.describe()
```

```
Out[6]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.762398	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype  
---  --
0   sepal length (cm)      158 non-null   float64
1   sepal width (cm)       158 non-null   float64
2   petal length (cm)      158 non-null   float64
3   petal width (cm)       158 non-null   float64
4   target                 158 non-null   int32   
types: float64(4), int32(1)
memory usage: 5.4 KB
```

```
In [8]: df.corr()
```

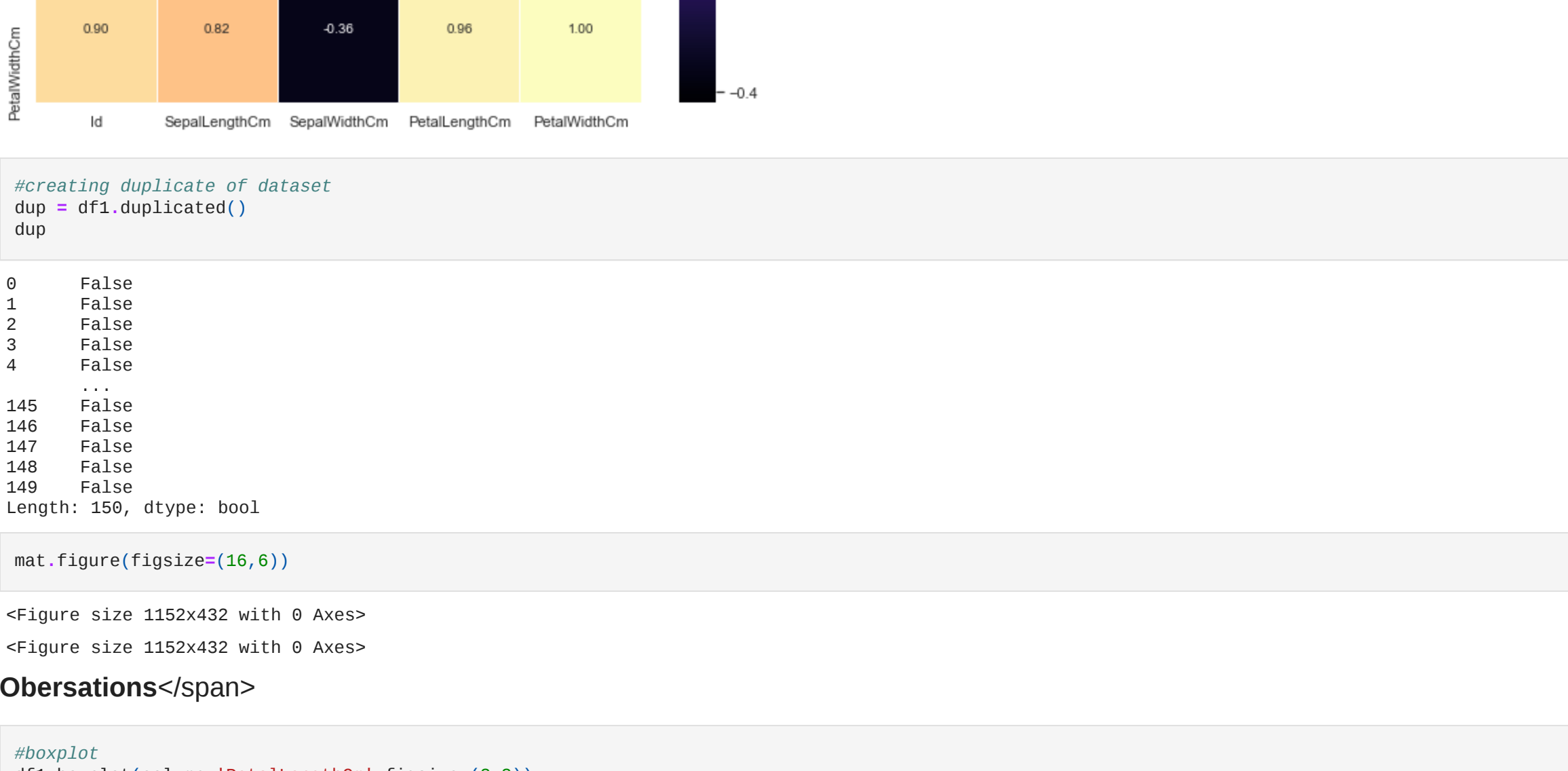
```
Out[8]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
sepal length (cm)	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal length (cm)	0.871754	-0.428440	1.000000	0.962865	0.949035
petal width (cm)	0.817941	-0.366126	0.962865	1.000000	0.956547
target	0.782561	-0.426658	0.949035	0.956547	1.000000

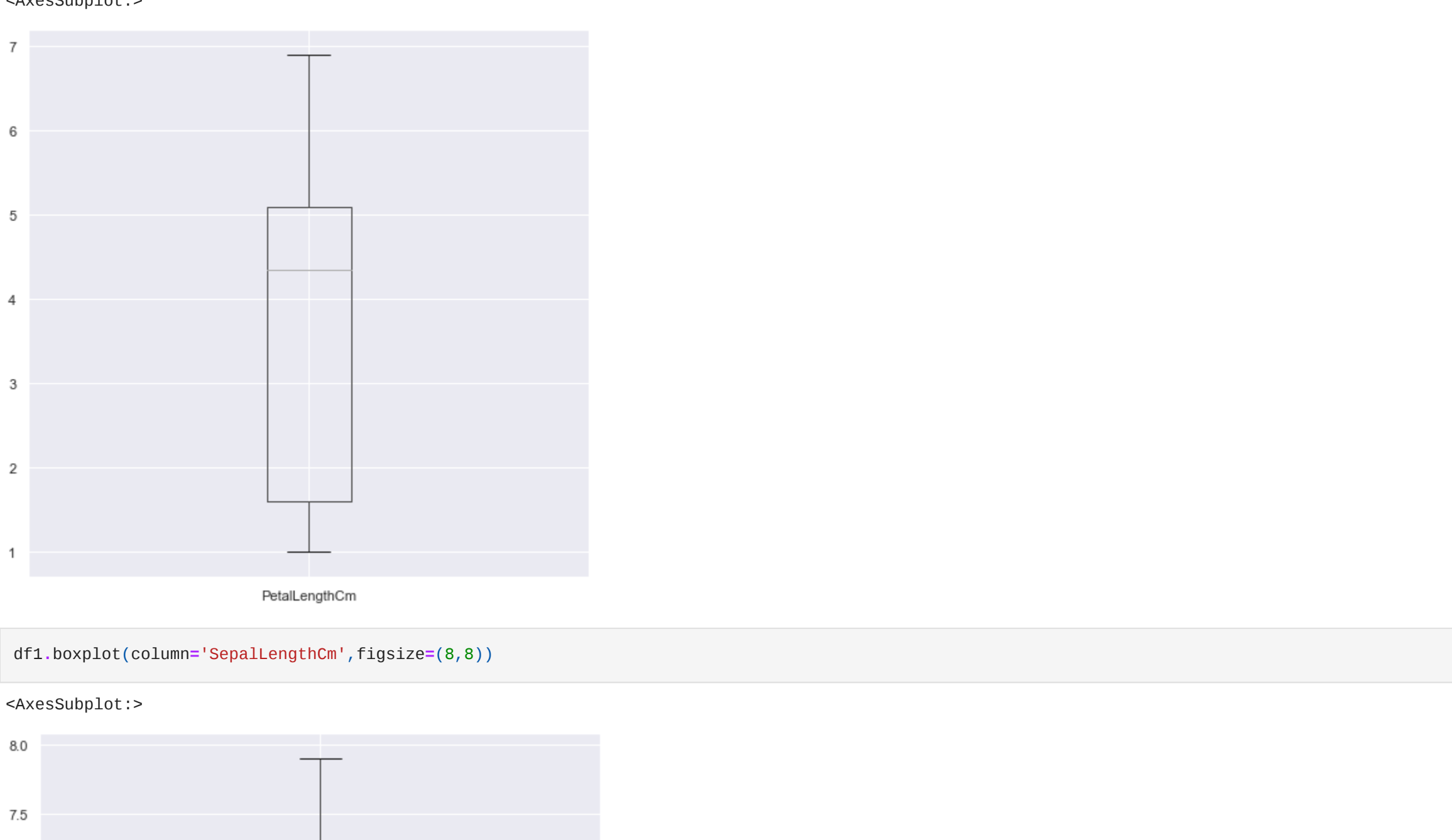
```
In [9]: #showing only target variable is the dependable variable
print(df['target'])
```

```
0    0
1    0
2    0
3    0
4    0
...
145  2
146  2
147  2
148  2
149  2
dtype: int32
Name: target, Length: 150, dtype: int32
```

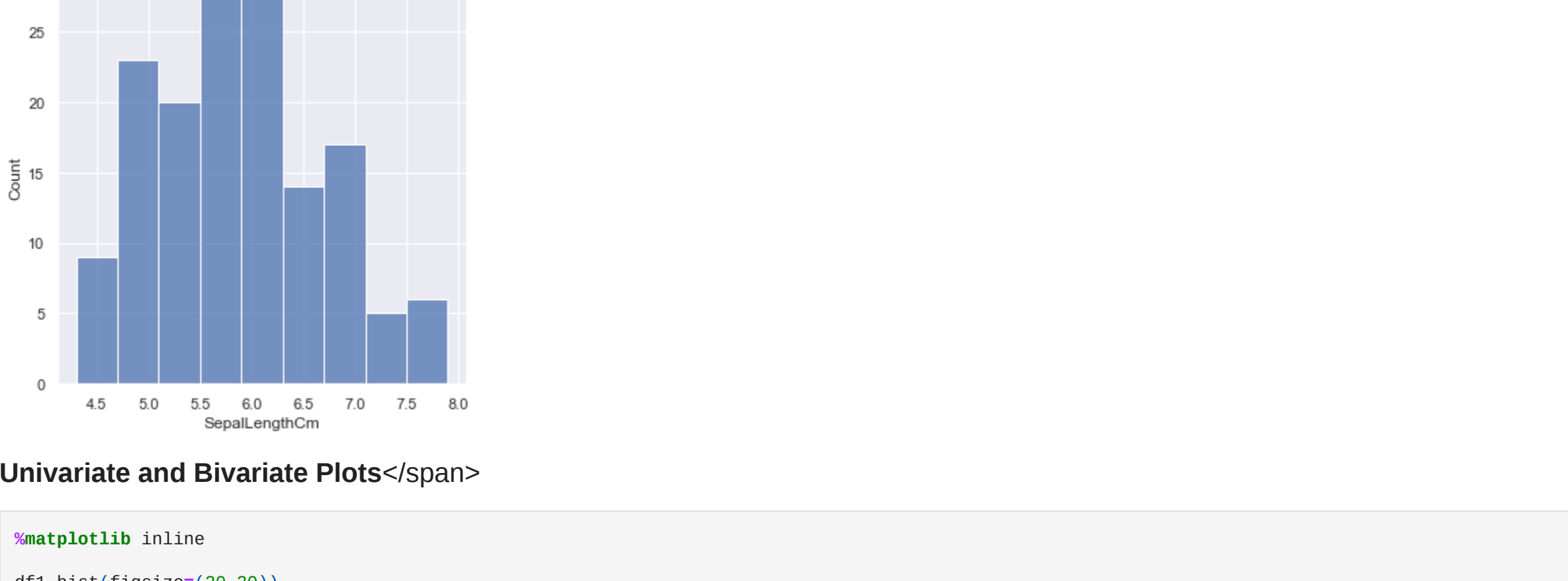
### Data Visualization</span>



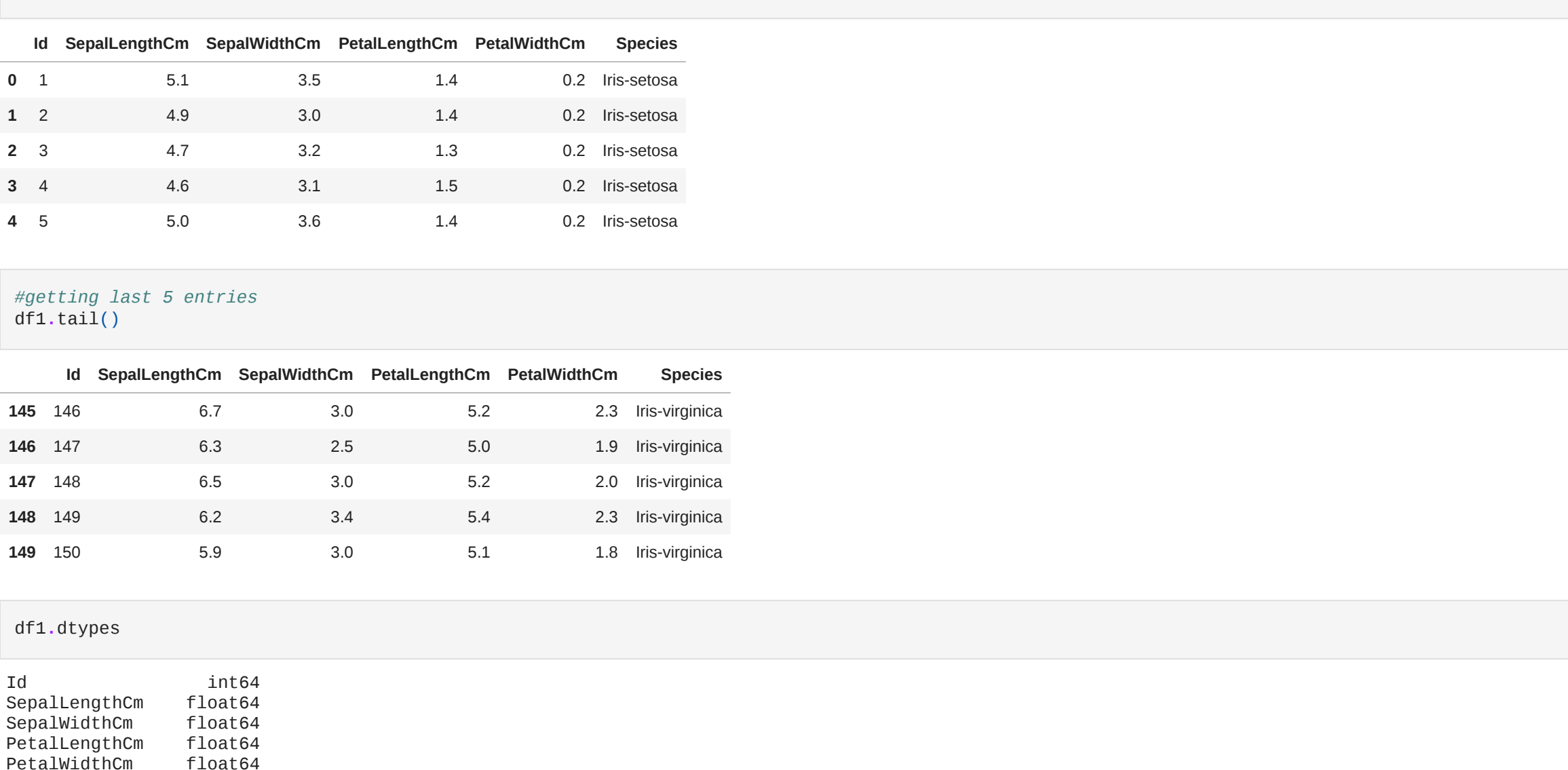
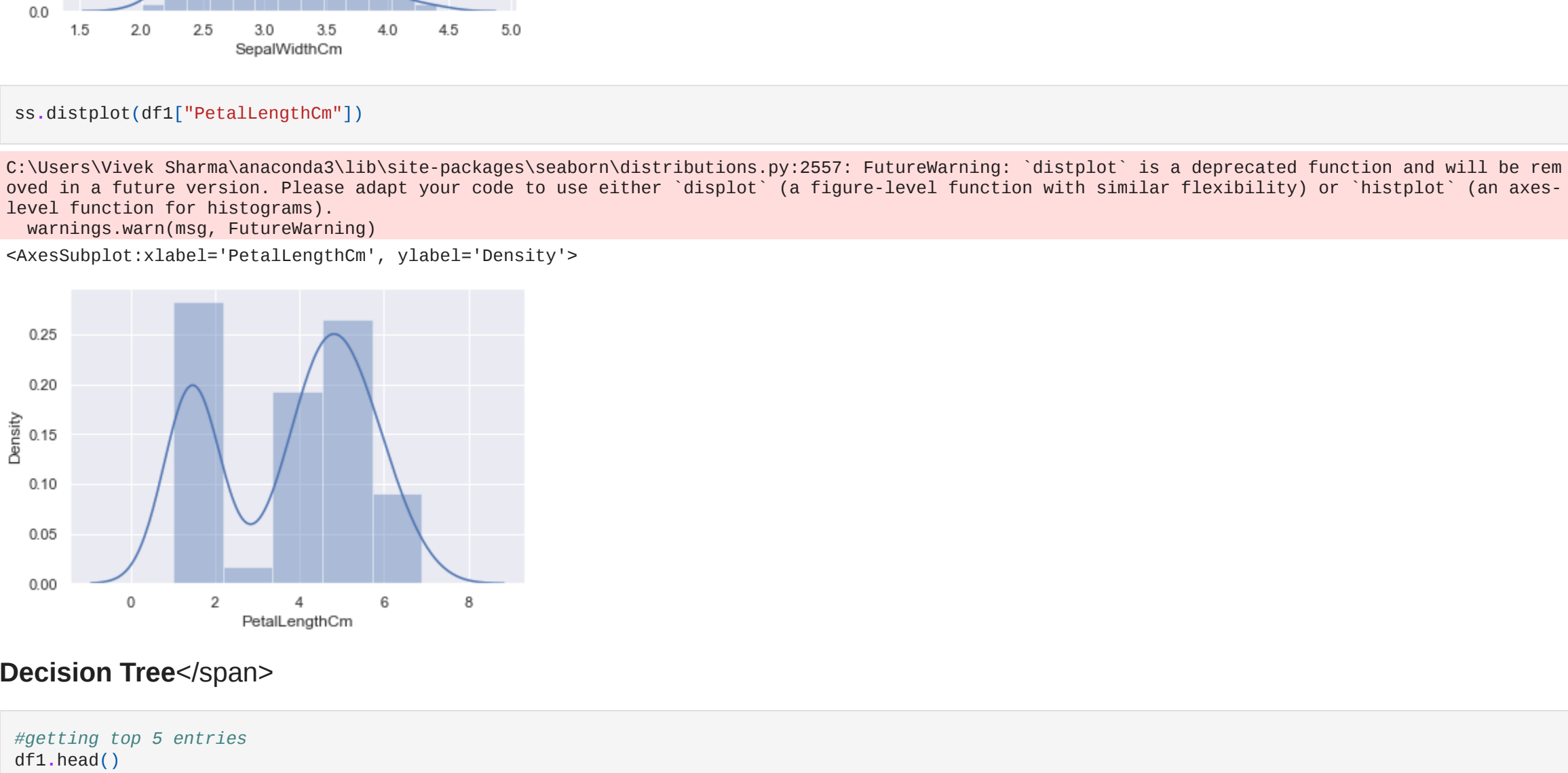
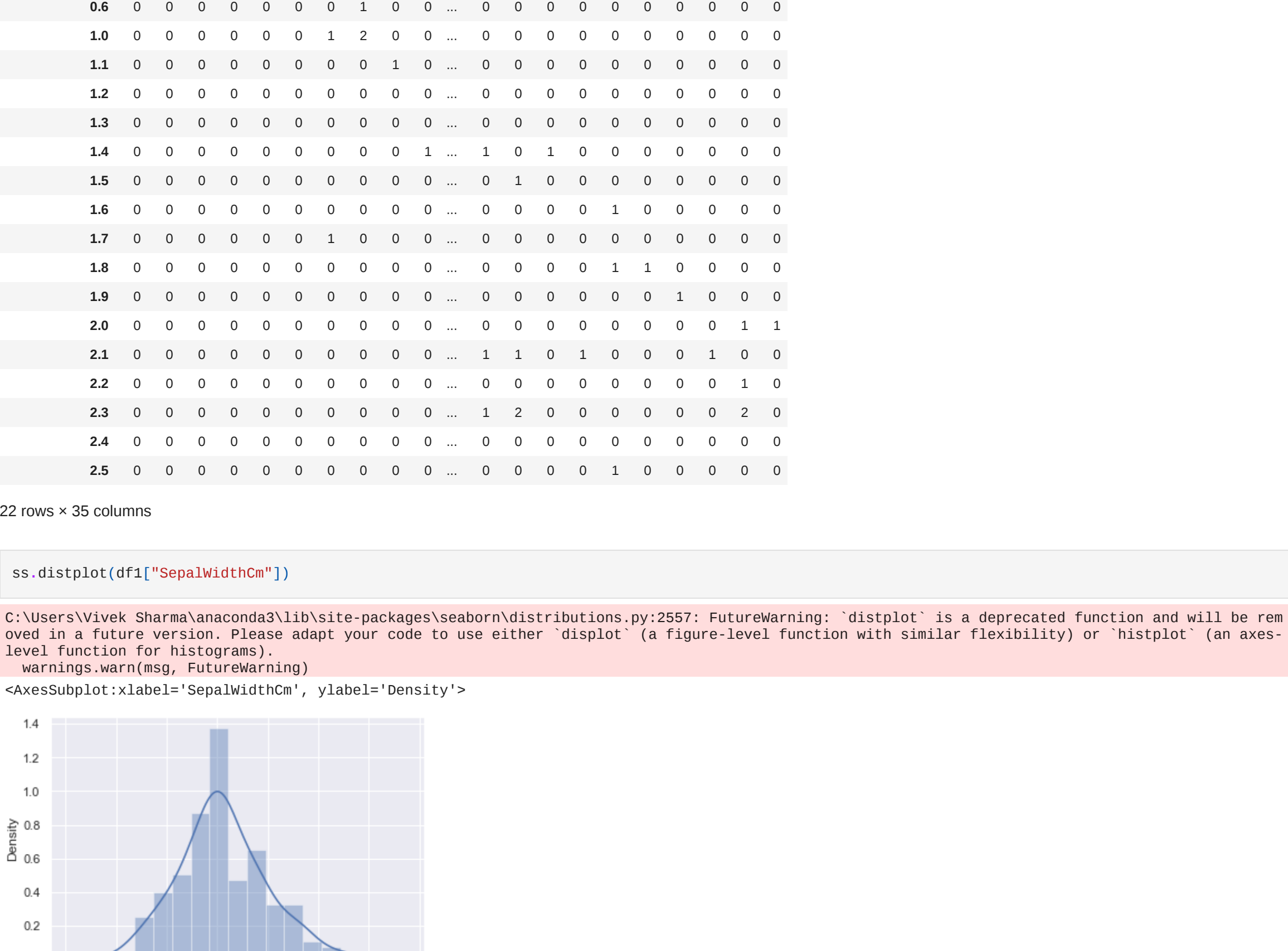
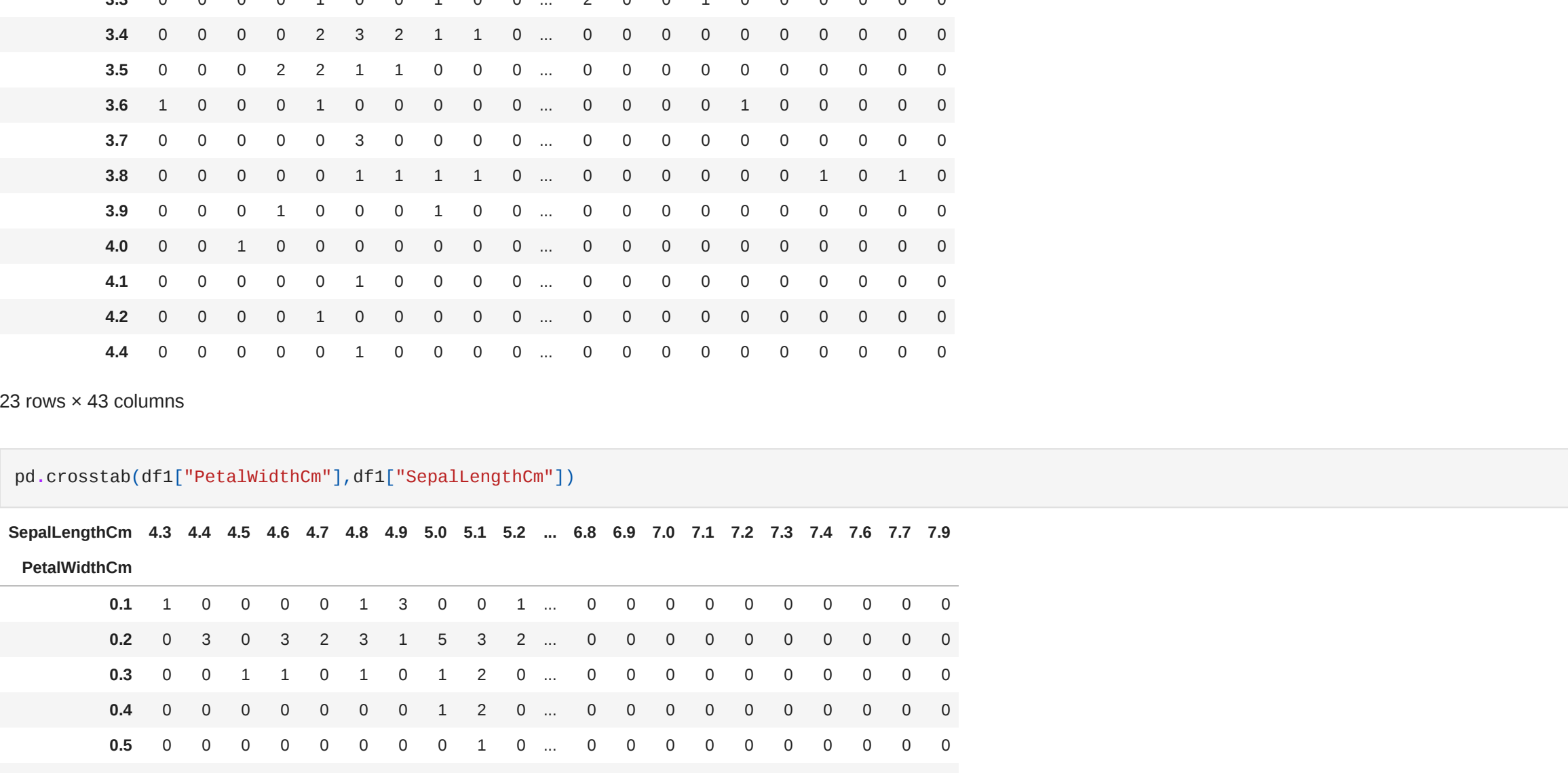
### Observations</span>



### conclusion</span>



### Univariate and Bivariate Plots</span>



### Decision Tree</span>

